store
67 25/9/24

{ Program - 1 } (hello world)

```
public class hello world {
    public static void main (String[] args ) {
        System.out.println ("hello world");
    }
}
```

Output

hello world

{ Program 2 (prime numbers) }

```
import java.util*
public class prime {
    public static void main (String[] args ) {
        int num = 29;
        int cnt = 0;
        for (int i = 2; i <= num/2; i++){
        if (num % i == 0) {
        System.out.println ("Not prime");
        cnt++;
        break;}}
        if (cnt == 0){
        System.out.println ("prime");}}}
```

Output:

prime

{Program - 3 fibbonacci series}

```
public class fibonaceiSeries{
    public static void main (string[] args){
        int n = 10;
        int y = 0;
        int x = 1;

    for (int i = 1; i <= n; ++i){

        int z = y + x;
            y = x;
            x = z;
        }
    System.out.println (z);

    }
    }
```

Output:

0 1 1 2 3 5 8

{Program - 4 triangle scalene, isoceles, equilateral}

```
public class triangle {
    public static void main (String [] args ) {
        int a = 10;
        int b = 20;
        int c = 30;
    if (a==b && b==c && c==a) {
        System.out.println ("equilateral triangle");
    }
    else if [redacted] (a==b || b==c ||a=
        [redacted]
        System.out.println ("not equilateral it i
    isocelles[redacted]);
        }
    else {
        System.out.println ("scalenen");}
    }}
```

Output

scalene

{ program - 5 simple interest } .

```
public class SI {
    public static void main (String [] args )
    {
        int p=1000 , r = 5, t = 3, SI;
        SI = (p* r * t )/100 ;
        System.out.print (SI);
    }
}
```

output: 150

{ program - 6 - swap 2 numbers }

```
public class swap {
  public static void main (String [] args)
  {
     int a = 5, b = 9;
     int c;
     a = b;
     b = c;
     System.out.println ("a=" + a + "b=" + b);
  }
}
```

a = 9 b = 5

2024

Week-1

Develop a java program that prints all real solution to the quadratic equation $ax^2 + bx + c = 0$. Read in a, b, c and use the quadractic formula. If the descriminate $b^2 - 4ac$ is negative, display message starting that there are no real solution.

```
import java.util.*;
class Quad {
Scanner sc = new Scanner(System.in);
int a, b, c, d;
double r1, r2, dsq;

void input() {
System.out.println("Enter coefficient a,b,c:");
a = sc.nextInt();
b = sc.nextInt();
c = sc.nextInt();
}
void calc() {
d = b*b - 4*a*c;
if (d==0) {
r1 = -b/(2.0*a)
System.out.println("Roots are real and distinct");
System.out.println("Root 1 = "+r1+" + "+r2+"i"+"\n"+
                   "Root 2 = "+r1+" - "+r2+"i");
}}}
class Quadratic {
public static void main(String[] args) {
Quad quad = new Quad();
quad.input();
quad.calc();
}
}
```

```
else if (d > 0) {
d_sq = Math.sqrt (d);
r1 =  (-b + d_sq ) / (2.0 * a );
r2 = (-b + d_sq ) / (2.0 * a );
System.out.println ("Roots are real & distinct");
System.out.println ("Root 1 = " + r1 " \n Root 2 = "
}
else {
d_sq = Math. sqrt (-d);
r1 = -b / (2.0 * a);
r2 = d_sq / (2.0 * a );
System.out. println ( " Roots are imaginary ")
System.out. println ( "Root 1 = " + r1 + " " + r2 + "i
                    \n Root 2 = " + r1 + " - " + r
}
}
```

Output:

enter coefficient a, b, c

1

2

2

-4

Roots are imaginary

Root 1 = -1.0 + 1.2456 i

Root 2 = -1.0 + 1.2456 i

enter coefficient a, b, c :

2

3

4

-23

Roots are imaginary

Root 1 = -1.0 + 1.41421i

Root 2 = -1.0 + 1.41421i

enter coefficient a, b, c :

1

2

2

-4

Roots are imaginary

Roots 1 = -1.0 + 1.9895

Roots 2 = -1.0 + 1.9895

o/p seen

9/10/24

```
import java.util.*scanner;
class student
{
string name;
string usn;
int credits[] = new int[5];
int marks[] = new int[5]
double sgpa = 0.0;
double cgpa = 0;
int grade[] = new int[5];
double calculate (int m[], int c[])
{
int j;
double sum = 0.0;
int div = 0;
for (j=0; j<5; j++)
{
if (m[j] != 100)
{
grade[j] = 10;
}
div = credits[j] + div;
sum = sum + (grade[j] * credits[j]);
System.out.println ("grade of subject " + (j+1) + " :
                    grade[j]));
}
sgpa = sum / div;
System.out.println ("SGPA " + sgpa);
return sgpa;
}
double calcgpa (double sgpa1, double sgpa2)
{
cgpa = (sgpa1 + sgpa2) / 2
return cgpa; }
```

```
Void input ()
{
Scanner sc = new scanner (System.in)
System.out.println ("Now enter subject credits for semester 4);
int i;
for (i= 0; i< 5; i++ )
{
credits [i] = sc. nextInt () ;
}
System. out. println ("Now enter subject marks for semester :4)
for (i=0; i<5 ; i++)
{
mark [i] = sc. nextInt ();
}
}
Public static Void main (string args [])
{
Scanner sc1 = new Scanner (System. in );
System. out. println ("Enter number of students : " );
int n= sc1. nextInt ();
Student obj [] = new student [n];
int k;
for (k=0 ; k< n; k++ )
{
obj [k]= new Student ();
System. out. println ("enter student name : ");
name = sc1. nextLine ();
obj [k] = input ();
System. out . println ("semester 1 :");
double result = obj[k] . calculate (obj [k]. marks,
obj[k]. credits );
System. out. println ("1st semester SGPA for " +
obj [k]. name+" (" + obj [k]. usn
```

```
System.out.println ("semester 2 ");
obj [k]. input ();
double result 2 = obj [k]. calculate (obj [k]. marks,
  obj [k]. credits);
System. out. println ("2nd semester SGPA for " + "
                obj [k]. usn + ") is: " + re
System. out. println (" CGPA for 1st year is:
                " + obj [k]. ~~usn~~ ") is, result
                cal cgpa (result
  }
 }
}.
```

```
Obj[k] = new studen ();
System. out. println ("Enter student name: ");
String name = sc1. next();
System. out. println ("Enter student name USN: ");
String usn = sc1. next();
```

seen
execute

Output:

Enter name:

abhi

Enter USN:

123456

Now enter subject credits for semester

3

3

3

3

3

3

3

3

Now enter marks for each semester

100

100

100

100

100

100

100

100

Semester 1

Grade for subject 1 : 10

Grade for subject 2 : 10

Grade for subject 3 : 10

Grade for subject 4 : 10

Grade for subject 5 : 10

Grade for subject 6 : 10

Grade for subject 7 : 10

Grade for subject 8 : 10

1st Semester SGPA is 10.
Semester 2.
Now enter subjects credits for semester:
3
3
3
3
3
3
3
3
3
Now enter marks for each semester
100
100
100
100
100
100
100
100
100
Semester 2
Grade for subject 1 : 10
Grade for subject 2 : 10
Grade for subject 3 : 10
Grade for subject 4 : 10
Grade for Subject 5 : 10
Grade for subject 6 : 10
Grade for Subject 7 : 10
Grade for subject 8 : 10

2nd Semester SGPA is 10.
CGPA for 1st year is 10.00 :

Program — 3

Create a class Book which contains 4 member, name, author, price, num_pages. Include a constructor to set the values for the members. Include methods to set & get the details of the object. Include a to string() method that could display the complete details of the book

```
import java.util.scanner;
class Book {
  private string name;
  private string author;
  private double price;
  private int numPages;

public Book (string name, string author, double
                price, int numPages) {

    this.name = name;
    this.author = author;
    this.price = price;
    this.numPages = numPages;
  }

Public void setName (string name) {
      this.name = name;
  }
Public void setAuthor (string author) {
      this.author = author;
  }
Public void setNumPages (int numPages) {
      this.numPages = numpages;
  }
Public void setPrice (double price) {
  }    this.price = price;
```

```
public String getName() {
    return Name; }
public String getAuthor() {
    return author ; }
public double getPrice() {
    return price; }
public int getNumPages () {
    return numPages; }
public string to string () { return " book Name : " +
 name + " Author " + author + " Price :" + price
 + " Number of pages " + numpages;
 }
}

public class BookMain {
    public static void main (String [] args ) {
        Scanner scanner = new scanner (System.in);
        System.out.println (" Enter the number of boo

        int n = Scanner.nextInt ();
        Scanner.nextLine ();

        Book[] books = new Book[n];

        for (int i=0 ; i<n ; i++ ) {

        System.out.print[" Name "];
        String name = Scanner.nextLine ();
        System.out.print ("Author: ");
        String author = Scanner.nextLine ();
        System.out.print ("Price: ");
        double price = Scanner.nextDouble ();
```

```
System.out.print("Number of pages: ");
int numPages = scanner.nextInt();
scanner.nextLine();

book[i] = new Book(name, author, price,
numPages);
}
System.out.println("\n book details:\n");
for(Book book: books){
    System.out.println(book);
}
scanner.close();
}
}
```

Out put :-

Enter the number of books : 3
Enter details for book 1:
Name : a
author : a
Price 1
Number of pages : 1
Enter details for book 2:
Name: b
Author: b
Price : 2
Number of pages: 2
Enter details for book 3:
Name : c
author : c
Price: 3
Number of pages : 3

abhinav.c IBM23CS008

book Name a , Author a , Price 1, Number of page
book Name b, Author d, Price 2, Number of pa
book Name d, Author d, Price 3, Number of p

2021

Prgrm – 4

store 67

2) Develop a java program to create an abstract class named shape that contains two integers & an empty m named printArea(). Provide 3 classes named Rec Triangle, Circle such that each one of the classes c only method printArea() that prints the area of given shape.

```
abstract class Shape {
    protected int dimension1;
    protected int d2;
    public abstract void printArea(); }

class Rectangle extends Shape {
    public Rectangle(int length, int width) {
        this.d1 = length;
        this.d2 = width;
    }

    Public void printArea() {
        int area = d1 * d2
        System.out.println("Area of rectangle " + are
    }
}

class Triangle extends Shape {
    public Triangle(int base, int height) {
        this.d1 = base;
        this.d2 = height;
    }
```

```
public void printArea(){
    double area = 0.5 * d1 * d2;
    System.out.print("Area of tri " + area);
}
}

class Circle extends Shape {
    public Circle(int radius){
        this.d1 = radius;
    }

    public void printArea(){
        double Area = Math.PI * d1 * d1;
        System.out.println("Area of circle" + area);
    }
}

public class ShapeTest {
    public static void main(String[] args){
        Shape rectangle = new Rectangle(5, 3);
        rectangle.printArea();

        Shape triangle = new Triangle(4, 6)
        triangle.printArea();

        Shape circle = new Circle(7);
        circle.printArea();
    }
}
```

Output :

Area of Rectangle : 15

Area of Triangle : 12

Area of circle : 153.938

Program - 5

Write a java code & make 2 cust name, acc no bank, Saving account and current account ask for deposit, withdraw, display balance.

```
import java.util.Scanner
class Account {
protected String customerName;
protected String accountNumber;
protected double balance;

public Account (String customerName, String accountNumber,
                double initialBalance) {
  this.customerName = customerName;
  this.accountNumber = accountNumber;
  this.balance = initialBalance;
  }

Public void deposit (double amount) {
   balance += amount;
  System.out.println ("deposited :" +amount);
}
Public void displayBalance() {
  balance  System.out.println ("current balance" +
                balance);
  }
Public void withdraw( double amount ) {
if (amount > balance) {
System.out.println ("insufficient balance");
}
else {
balance == ammount && balance > amount
  System.out.println ("
```

```
class SavAcc extends Accounts {
private double intrestRate ;
public savAcct (string customerName, String accountN
                double initialBalance, double intrestR
        this.Rate = intrestRate;
    }
Public void computeAndDepositInterest (int years){
    double interest = balance * Math.pow((1+intrestRate/
                    years) - balance;
        deposite (interest);
    System.out.println ( "intrest for" + years + "years
                + interest);
    } }
class curr Acct extends Account {
private double minimum balance;
private double service charge;
public curr Acct (string customerName, String accountNum
                double minimumBalance, double serivce
Super (customers Name, accountNumber, Initial Bal
    this. minimumBalance = minimumBalance;
    this. service charge = Service charge ;
}
@ Override
public void withdraw (double amount ) {
    if (amount > balance ){
        System.out.println ("insufficient balance!");
else {
balance = amount;
System.out.println ("withdrew" + amount);
check minimum balance ();
}
}
```

```
private void checkMinimumBalance(){
    if (balance < minimumbalance ){
        balance -= serviceCharge;
System.out.println("minimum balance not maintained.
                    Service charge of"+ servicecharge +
                    applied");
    }}}
public class Bank{
public static void main(String[] args){
Scanner sc = new Scanner(System.in);
    Account account = null;
System.out.println("Enter customer name:");
String name = sc.nextLine();
System.out.println("Enter acc no");
Stric accountNumber = sc.nextLine();
    System.out.println("choose account type (1 for saving,
int accountType = sc.nextInt();          2 for current ):")
if (accountType == 1){
    System.out.print("Enter initial balance");
    double initialBalance = sc.nextDouble();
    System.out.print("Enter intrest rate");
    double intrestRate = sc.nextDouble();
    account = new SavAcc(name, accountNumber, initial
} else if (accountType == 2){ balance, intrestRate).
    System.out.print("Enter initial balance");
    double initialBalance = sc.nextDouble();
    System.out.print("Ents minimum balance");
    double minimumBalance = sc.nextDouble();
    System.out.print("Entr service charge");
    double serviceCharge = sc.nextDouble();
    account = new CurrAcct(name, accountNuber,
                initial balance, minimumbalance, serviceCh
```

```
} else { System.out.println ("Invalid account type");
return;
}
int choice;
do { System.out.println ("In Menu");
System.out.println ("1 Deposit");
System.out.println ("2 Display balance");
System.out.println ("3 withdraw");
System.out.println ("4 compute & deposite Intrest (saving only)"
System.out.println ("5. Exit");
System.out.println ("Enter your choice");
choice = sc.nextInt();
Switch (choice) {
case 1:
System.out.printf: "Enter amount to deposit:");
double depositAmount = sc.nextDouble();
account.withdraw (withdrawAmount);
break;
case 2:
account.displayBalance();
break;
case 3: System.out.println ("Enter amount to be withdrawn"
double withdrawAmount = sc.nextDouble();
account.withdraw (withdrawAmount);
break;
case 4: if (account instance of SavAcct) { Sy.o.p ("Enter numb
of years for interest calculation"); int years = sc.nextInt();
((SavAcct) account).computeandDepositeIntrest (years); } else
System.out.println ("This option is only available for savi
account"); } break;
case 5: System.out.println ("thank you for using our servi
break;
default:
System.out.println (invalid choice! Please select a valid
option"); break;
}
while (choice != 5);
sc.close(); } }
```

«Lab-6»

1) Create a package CIE which has 2 classes - student & internals. The class Student has members like USN, name, sem. The class internals derived from student has an array that stores internal marks scored in 5 courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of the student. Import the 2 package in a file that declares the final marks of n student in all 5 courses

```
Package CIE;
public class student{
    protected string name;
    protected int[] marks;
public student (String name){
    this.name=name;
    this.marks = new int[5]; }
public string getName(){
        return name;
    }
Public void setMarks (int[] marks)
    this.marks = marks;
}
public int[] get marks () {
    return marks;
    }
}
```

```
Package CIE;
public class Internal extends Student {
    private int[] internalMarks;
Public Internal (String name, int[] internalMarks){
 Super(name);
 this.internalmarks = internalMarks;
 this.setMarks(internalMarks);
}
public int[] getInternalMarks(){
    return internalmarks;
}
public void setInternalMarks(int[]internalmarks) {
  this.internalmarks = internalMarks;
 this.setmarks = (internalMarks);
 }

Package SEE;
import CIE.student;
public class External extends student {
private int[] external marks;
public External (String name, int[] externalMarks) {
 Super(name)
 this.external marks = external marks;
 this.setmarks (external Marks);
}
Public int[] getExternalMarks (){
    return external Marks;
}
Public void setExternalMarks (int[] external Marks)
  this.externalmarks = external marks;
  this.setmarks (external Marks);
```

```
import CIE.internal;
import SEE.external;
import java.util.scanner;
Public Class Main {
 public static void main (String [] args ) {
    Scanner Sc = new scanner (System.in );
 System.out.println ("Enter no. of students");
    int n = SC.nextInt();
    Sc.nextline();
Internal[] internal students = new Internal[n]
External[] external students = new External [n] ;

for (int i=0 ; i<n; i++ ) {
  System.out.println ("Enter name of s" +(i+1)+":"
  String.name = sc.next line ();
  system.out.println ("Enter 5 course " +name +: ");
  int[] internal marks = new int [5];
  for ( int j =0; j<5 ; j++ ) {
    internal marks[j] = sc.next Int();
}
Sc next line ();

System.out-println ("Enter external marks of
                    (5 courses) for +name +:");
int[] external = new int[5];
for (int j =0; j<5; j++ ){
    external marks [j] = sc.next Int();
    }
Sc.next line ();
```

```
internal students s[i] = new internal (name, internal Mark
external students [i] = new External (name, externalmark
}
System.out.println("\n Final Marks for all students :");
for (i=0 ; i<n; i++) {
int [] internal marks = internal students [i].getmarks(
int [] external marks = externalstudents [i].getmarks
System.out.println("\n student " + internalstudents [i].
                    getName() );
System.out.print(" Internal marks : ");
for (int mark: internal marks ) {
   System.out.println(marks + " ");
}
System.out.print("\n External marks : ");
for (int mark: external marks ) {
   System.out.print(mark + " ");
}

System.out.print("\n Final marks :");
for (int j = 0 ; j< 5 ; j++ ) {
   int final mark = internal marks [j] + external marks
   System.out.print(final mark + " ");      [j] ;
}
System.out.println();
}
sc.close();
}
}.
```

O/P :-

Enter number of students: 1

Enter the name of student 1: abhinav

Enter internal marks (5 courses) for abhinav:

1

1

1

1

1

Enter external marks (5 courses) for abhinav:

2

2

2

2

2

Final marks for all students:

Student: abhinav.

internal marks : 1 1 1 1 1

external marks : 2 2 2 2 2

Final marks : 3 3 3 3 3

O/P for program 5:-

Welcome to the Bank!
Enter customer name: A
Enter account number: 10
choose account type (1 for saving, 2 for current): 1
Enter initial balance: 10
Enter intrest rate: 100
Menu:
1. Deposit
2. Display balance
3. withdraw
4. compute & deposite Intrest (saving only).
5. Exit.
Enter your choice: 1
Enter amout to deposite: 10
current balance: 10.
Menu:
1. Deposit
2. display balance
3. withdraw
4. compute & deposite Intrest (saving only)
5. Exit
Enter choice: 3
Enter amount to withdraw 9
with drew 9
1. Deposit
2. display balance
3. withdraw
4. compute & deposite Intrest (saving only)
5 exit
Enter no. of years for intrest calculation: 100.
deposited 2.533
Intrest for ... deposited : 2.535 301200

Q) Define a subclass of exception (which is, of course, a subclass)
WAP that demonstrate handling of exception in inheritance tree. Create base class father & subclass son which extends base class. In fathers class implement a constructor which takes age & throws exception wrong age when input age is less than 0. In sons class implement constructor that uses both fathers & sons age & throws an exception if son ≥ father age

```
class Father {
        String name;
```

```
import .java.util.scanner;
class NegativeAgeError extends Exception {
int a;
public NegativeAgeError(int a) {
this.a = a;
}
public string to string () {
  return "Negative Age :"+a;
}
}
class InvalidAgeError extends Exeption {
int a, b;
public InvalidAgeError(int a, int b) {
  this.a = a;
  this.b = b; }
public string tostring(){
    return "Invalid Age :" + a + "is not less than "+b;
  }
  }
class Father {
 string name;
  int Age;
  Father (string name, int age ){
    try {
      if (age % < 0) {
          throw new NegativeAgeError(age);
          }
          this.name = name;
          this.age = age;
          }
          catch (Negetaive AgeError e) {
            this.age = 20.
          System.out.println(e);
} } }
```

```
class Son extends Father {
String sonName;
int sonAge;
Son (String sonName, int sonAge, String fatherName, int
    fatherAge) {
      super (fathenName, fatherAge);
      this.sonName = sonName;
      try {
        if (sonAge < 0) {
          throw new NegativeAgeError(sonAge);
        }
        if (sonAge >= this.age) {
          throw new InvalidAgeError(sonAge this.age);
        this.sonAge = sonAge;
        }
      Catch (NegativeAgeerror e) {
        this.sonAge = 20;
        System.out.println(e);
      }
      catch (invalidAgeError e) {
        this.sonAge = 10;
        System.out.println(e)
      }
}
}
```

```
class Exceptions {
public static void main(String[] args){
    Scanner scanner = new Scanner(System.in);

    System.out.print("Sons Name");
    String sonName1 = scanner.nextLine();
    System.out.print("Son Age");
    int sonAge = scanner.nextInt();
    scanner.nextLine();
    System.out.println("Father Name");
    String fatherName = scanner.nextLine();
    scanner.nextLine();
    Son a1 = new Son(sonName1, sonAge1,
                     fatherName1, fatherAge1);

    System.out.println("Son Name:" + a1.sonName +
                       "Age:" + a1.sonAge);
    scanner.close();
}
}
```

Output:

Enter sons age: 10

Enter father age: 5

Age error

Enter sons age: -10

negative age error.

seen

20/11/24

Lab-8

write a program which creates 2 threads, one thread display "BMS college of Engineering" once every 10 seconds & another displaying "CSE" once every 2 seconds.

```
class BMS extends Thread {
    public void run () {
        while (true) {
            System.out.println ("BMS college of Engineering");
            try {
                Thread.sleep (10000);
            }
            catch (InterruptedException e) {
                System.out.println (e);
            }
        }
    }
}
class CSE extends Thread {
    public void run () {
        while (true) {
            System.out.println ("CSE ");
            try {
                Thread.sleep (2000);
            }
            catch (InterruptedException e)
            {
                System.out.println (e);
            }
        }
    }
}
```

```
public class CS_IS_ThreadProgram {
    public static void main(String[] args) {
        BMS bmsThread = new BMS();
        bmsThread.start();
        CSE cseThread = new CSE();
        cseThread.start();
    }
}
```

output:

```
BMS college of Engineering
CSE
CSE
CSE
CSE
CSE
BMS college of Engineering
CSE
CSE
CSE
CSE
CSE
```

7/11/24

Open ended java question:-

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

class SwingDemo{
  SwingDemo(){
  JFrame jfrm = new JFrame("Divide App");
  jfrm.setSize(275, 150);
  jfrm.setLayout(new FlowLayout());
  jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
  JLable jlab = new JLabel("Enter dividend & divisor");
  JTextField ajtf = new JTextField(8);
  JTextField ajtf = new JTextField(8);
  JButton = new button("Calculate");
  JLabel err = new JLabel();
  JLabel alab = new JLabel()
  JLabel blab = new JLabel();
  JLabel anslab = new JLabel();
  jfrm.add(err);
  jfrm.add(jlab);
  jfrm.add(ajtf);
  jfrm.add(bjtf);
  jfrm.add(button);
  jfrm.add(alab);
  jfrm.add(blab);
  jfrm.add(anslab);
```

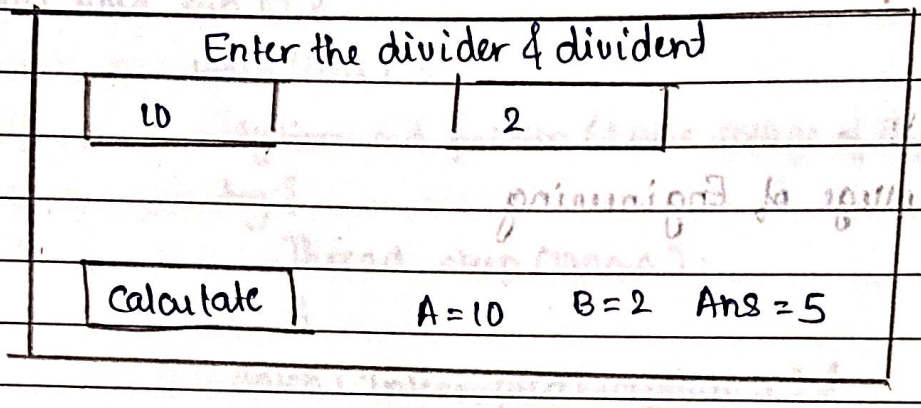
```
Action Listener l = new Action Listener(){
public void action Performed (Action event evt){
    System.out.println("Action event from a text field");
}};
abtb.addActionListener(l);
bjtf.addActionLisler(l);
button.addActionLislines (new ActionLisler(){
    public void actionPerformed (ActionEvent evt){
    try {
        int a = integer.parseInt(ajtf.getText());
        int b = integer.parseInt(bjtf.getText());
        int ans = a/b;
        alab.SetText("in A = "+a");
        blab.SetText("in B = "+b);
        anslab.setText("in Ans="+ans);
    }
    catch (NumberFormatException e){
    alab.setText("");
    blab.SetText("");
    anslab.setText("");
    err.setText("Enter only integer!");
    }
    catch (ArithmeticException e){
    alab.setText("");
    blab.setText("");
    anslab.setText("");
    err.setText("B should be NONZero!");
}}});
jfrm.setVisible(true);
}
```

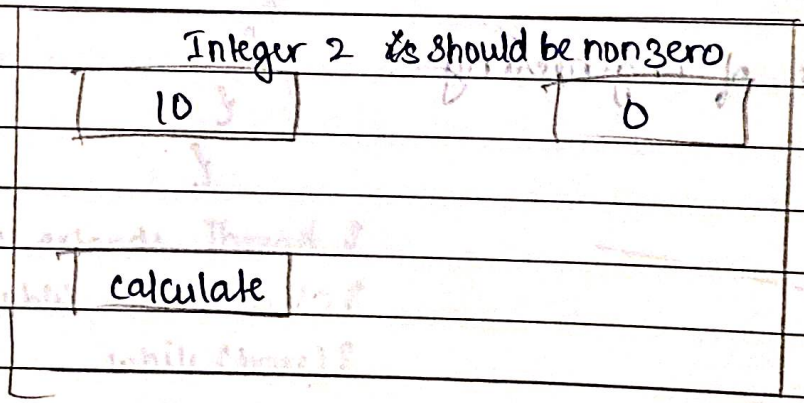
```
public static void main(String args[]){
    SwingUtilities.invokeLater(new Runnable(){
        public void run(){
            new SwingDemo();
        }
    });
}
```

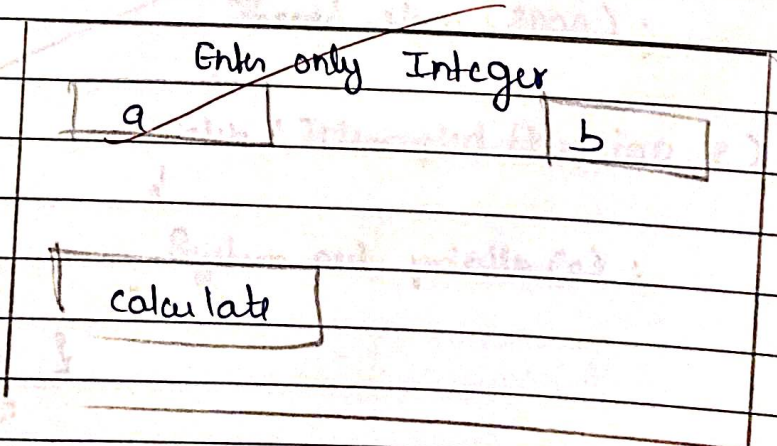
7/11/24

{program - 9}

Q) Write a program that creates a UI to perform integer div
The user enters 2 numbers in the text fields Num1 & N
The division of N1 & N2 is displayed in the Result fi
when divide button is clicked. if N1 or N2 were not int
would throw error if N2 is 0 throws error.







ipu:

```
class Q {
int n;
boolean value set = false;
Synchronized int get() {
    while(valueset) {
    try {
        System.out.println("In consumer waiting \n");
        wait();
    } catch (Interrupted Exception e) {
        System.out.println("interuplend Exception caught ");
    } }
    System.out.println("Got" + n);
    value set = false;
    System.out.println("in intimate procedure");
    notify();
    return n;
    }
Synchronized void put(int n) {
    while (value set) {
        try {
        System.out.println("in Procedure waiting \n");
        wait();
        } catch(interruptedException e) {
        system.out.println("interupted Exception caught
    } }
    this.n = n;
    value set = true
    System.out.println("Put" + n);
    System.out.println("intimate Consumer");
    }
}
```

```
class Procedure implements Runnable {
Qq;
Procedure (Qq) {
  this.q = q;
  new Thread (this, "Procedure").start();
}
Public void run() {
  int i = 0;
  while (i<15) {
  q.put (i++);
  try {
    Thread.sleep (500)
  } catch (InterruptedException e) {
  System.out.println ("Procedure interrupted");
}}}}
class Consumer implements Runnable
  Qq;
  Consumer (Qq) {
    this.q = q
    new Thread (this, "Consumer").start();
  }
  Public void Run() {
    while (true) {
    q.get();
    try {
      Thread.sleep (1000);
    Procedure
    } catch (InterruptedException e) {
      System.out.println ("Consumer");
}}}}
```

```
Public class Main {
    public static void main(String[]args){
    System.out.println("Abhinav 1BM23CS008");
    Qq q=new Q();
    new producer(q);
    new consumer(q);
    }
    }.
```

Computer science 5
Information Science 5.

Q) Demonstrate inter process communication & deadlock.

Output:

Put: 1

Got: 1

Put: 2

Got: 2

Put: 3

Got: 3

Put: 4

Got: 4

Put: 5

Got: 5

Deadlock:

```
class A {
synchronized void foo(B b){
    String name = Thread.currentThread().getName();
    System.out.println(name + "entered A.foo");

try {
  Thread.sleep(1000);
} catch (InterruptedException e){
System.out.println("A interrupted");
}
System.out.println("name + " trying to call B.last()");
b.last();
}
void last(){
  System.out.println("inside A.last");
}
}
class B {
synchronized void bar(A a){
    String name = Thread.currentThread().getName();
    System.out.println(name + "entered B.bar");
try {
   thread.sleep(1000);
} catch (InterruptedException e){
  System.out.println("B interrupted");
}
System.out.println(name + "trying to call
  A.last()");
  a.last();
}
void last(){
  System.out.println("inside B.last");
}}
```

```
class Deadlock implements Runnable{
    A a = new A();
    B b = new B();
Deadlock(){
    Thread.currentThread().setName("Main Thread");
    Thread t = new Thread(this, "RacingThread");
    t.start();
    a.foo(b);
    System.out.println("Back in main thread");
}
Public void Run(){
    b.bar(a);
    System.out.println("Back in other thread");
}
Public static void main(String args[]){
    new Deadlock();
}
}
```

Output

MainThread entered A.foo
RacingThread entered B.bar
Main Thread trying to call B.last()
Inside B.last
Back in main thread
Racind thread trying to call A.last()
Inside A.last
Back in other thread